CLAIMS

what is claimed is:

1. A method for performing design rule checking on an electronic design, the method comprising:

receiving a netlist of the electronic design, said netlist having a plurality of nodes;

extracting information from each of the plurality of nodes in the netlist, whereby the extracted information is available to a set of predefined rules;

applying the set of predefined rules to the previously extracted information of the plurality of nodes in the netlist; and

determining whether any of the predefined rules have been violated by any of the plurality of nodes, wherein the applying operation is performed by a rule checking routine, which checks the electronic design.

2. A method as recited in claim 1, wherein the applying operation comprises:

recognizing whether any of the predefined rules require extracted information from a neighboring node, the neighboring node being different from a node currently under consideration; and

identifying the neighboring node for comparing under the rule checking routine if any of the predefined rules require extracted information from the neighboring node, wherein the recognizing and identifying operations are collectively performed by a traverser routine.

3. A method as recited in claim 1, wherein the recited operations are each done automatically, without user intervention.

4. A method as recited in claim 1, wherein the extracting information operation is done only once for each node in the netlist.

5. A method as recited in claim 1, wherein the extracting information operation comprises determining what information is required to apply the predefined rules to the plurality of nodes.

6. A method as recited in claim 1, wherein the extracted information operation comprises one or more properties selected from the group consisting of fanin count, fanout count, atom type, and port source atom type.

7. A method as recited in claim 1, further comprising:
receiving the set of predefined rules.

8. A method as recited in claim 7, wherein the set of predefined rules comprises at least one hardcoded rule, which was predefined without any user intervention.

9. A method as recited in claim 7, wherein the set of predefined rules comprises at least one hardcoded rule, which has been modified in accordance with user instructions.

10. A method as recited in claim 7, wherein each predefined rule conforms to a single specified format.

11. A method as recited in claim 10, wherein the format is a hierarchical arrangement comprising a rule -> a sub-rule -> a basic rule.

12. A method as recited in claim 1, wherein the applying the set of predefined rules to the previously extracted information of the plurality of nodes in the netlist comprises:
applying a plurality of sub-rules related by a logical operator.

13. A method as recited in claim 12, wherein the logical operator is selected from the group consisting of AND, NAND, NOR, XOR, and OR.

14. A method as recited in claim 1, wherein the plurality of predefined rules is selected from the group consisting of electrical rules, connectivity rules, clock rules, timing closure rules, reset rules, and signal race rules.

15. A method as recited in claim 1, further comprising:
outputting a violation report, the report containing the identified violations.

16. An apparatus for performing design rule checking on an electronic design, the apparatus comprising:
a netlist creator that produces a netlist describing the functional representation of the electronic design across a plurality of nodes;
an information extractor that extracts information from the plurality of nodes in the netlist, the information being specific to each node; and

a rule checking engine that applies the set of predefined rules to the extracted information of the plurality of nodes in the netlist and that identifies whether any of the predefined rules has been violated by any portion of the electronic design.

17. An apparatus as recited in claim 16, further comprising:

a traverser engine that recognizes whether any of the predefined rules require extracted information from a neighboring node and further identifies the neighboring node for evaluating with the rule checking engine if any of the predefined rules require extracted information from the neighboring node, the neighboring node being separate from a node currently under comparison.

18. An apparatus as recited in claim 16, wherein the information extractor extracts information only once for each node in the netlist.

19. An apparatus as recited in claim 16, wherein the information extractor extracts information based on what is required to apply the predefined rules to the plurality of nodes.

20. An apparatus as recited in claim 16, wherein the information extractor extracts information that is selected from the group consisting of fanin count, fanout count, atom type, and port source atom type.

21. An apparatus as recited in claim 17, further comprising:

a rule creator for preparing a set of predefined rules.

22. An apparatus as recited in claim 21, wherein the rule creator is configured for selecting at least one hardcoded rule, the hardcoded rule having been predefined without any user intervention.

23. An apparatus as recited in claim 21, wherein the rule creator is configured for receiving at least one hardcoded rule and modifying the hardcoded rule in accordance with user instructions, the hardcoded rule having been predefined without any user intervention.

24. An apparatus as recited in claim 21, wherein the rule creator is configured for receiving the set of predefined rules from a user, and wherein each predefined rule conforms to a single specified format.

25. An apparatus as recited in claim 24, wherein the specified format is a hierarchical arrangement of a rule -> a sub-rule -> a basic rule.

26. An apparatus as recited in claim 25, wherein the specified format allows the rule to include a plurality of sub-rules related by one or more logical operators.

27. An apparatus as recited in claim 26, wherein one or more logical operators are selected from the group consisting of AND, NAND, NOR, XOR, and OR.

28. An apparatus as recited in claim 16, wherein the plurality of predefined rules is selected from the group consisting of electrical rules, connectivity rules, clock rules, timing closure rules, reset rules, and signal race rules.

29. An apparatus as recited in claim 17, further comprising:

a storage medium that stores the extracted information such that it is available to a set of predefined rules; and

30. An apparatus as recited in claim 17, further comprising:

an output device for outputting a violation report, the report containing the identified violations.

31. A computer program product comprising a machine readable medium on which is provided program instructions for performing design rule checking on an electronic design, the program instructions comprising:

instructions for receiving a netlist of the electronic design, said netlist having a plurality of nodes;

instructions for extracting information from each of the plurality of nodes in the netlist, whereby the extracted information is available to a set of predefined rules;

instructions for applying the set of predefined rules to the previously extracted information of the plurality of nodes in the netlist; and

instructions for determining whether any of the predefined rules have been violated by any of the plurality of nodes, wherein the applying operation is performed by a rule checking routine, which checks the electronic design.

32. A computer program product as recited in claim 31, further comprising:

instructions for recognizing whether any of the predefined rules require extracted information from a neighboring node, the neighboring node being different from a node currently under consideration; and

instructions for identifying the neighboring node for comparing under the rule checking routine if any of the predefined rules require extracted information from the neighboring node, wherein the recognizing and identifying operations are collectively performed by a traverser routine.

33. A computer program product as recited in claim 31, further comprising:
instructions for receiving the set of predefined rules.

34. A computer program product as recited in claim 31, further comprising:
instructions for outputting a violation report, the report containing the identified violations.

35. A method of performing design rule checking on a netlist representation of an electronic design, the method comprising:
(a) receiving a plurality of design rules specifying constraints on the properties of particular nodes in a netlist, wherein each rule specifies a logical combination of constraints on properties of particular nodes in the netlist, without including functionality for extracting said properties from the netlist;
(b) at a first node of the netlist, employing a generic routine to execute a first design rule from the plurality of design rules and determine whether properties of the first node violate the first design rule; and
(c) at the first node of the netlist, employing the generic routine to execute a second design rule from the plurality of design rules and determine whether properties of the first node violate the second design rule.

36. A method as recited in claim 35, further comprising:
extracting properties from the netlist and making them available for execution of the first and second design rules in (b) and (c).

37. A method as recited in claim 35, further comprising:
using the generic routine to execute a plurality of additional design rules at the first node.

38. A method as recited in claim 35, further comprising:
using the generic routine to execute the first design rule at a second node.

39. A method as recited in claim 35, further comprising:

using the generic routine to execute all design rules at first node, then using the generic routine to execute all design rules at a second node.

40. A method as recited in claim 35, further comprising:

using the generic routine to execute all design rules at a plurality of other nodes, then using the generic routine to execute all design rules at the plurality of other nodes.

41. A method as recited in claim 40, further comprising:

traversing to a neighboring node when any of the design rules require extracted properties from a neighboring node.

42. A method as recited in claim 35, wherein the extracted properties is selected from the group consisting of fanin count, fanout count, atom type, and port source atom type.

43. A method as recited in claim 35, wherein the plurality of design rules is selected from the group consisting of electrical rules, connectivity rules, clock rules, timing closure rules, reset rules, and signal race rules.

44. A method as recited in claim 35, wherein employing the generic routine comprises:

employing a logical operator for checking a plurality of sub-rules, the plurality of sub-rules being part of a corresponding design rule within the plurality of design rules.

45. A method as recited in claim 44, wherein the logical operator is selected from the group consisting of AND, NAND, NOR, XOR, and OR.

46. A computer program product comprising a machine readable medium on which is provided program instructions for performing design rule checking on a netlist representation of an electronic design, the program instructions comprising:

(a) instructions for receiving a plurality of design rules specifying constraints on the properties of particular nodes in a netlist, wherein each rule specifies a logical combination of constraints on properties of particular nodes in the netlist, without including functionality for extracting said properties from the netlist;

(b) instructions for employing a generic routine, at a first node of the netlist, to execute a first design rule from the plurality of design rules and determine whether properties of the first node violate the first design rule; and

(c) instructions for employing the generic routine, at the first node of the netlist, to execute a second design rule from the plurality of design rules and determine whether properties of the first node violate the second design rule.

47. A computer program product as recited in claim 46, further comprising:

instructions for extracting properties from the netlist and making them available for execution of the first and second design rules in (b) and (c).

48. A computer program product as recited in claim 46, further comprising:

instructions for using the generic routine to execute a plurality of additional design rules at the first node.

49. A computer program product as recited in claim 46, further comprising:

instructions for traversing to a neighboring node when the first design rule requires extracted properties from a neighboring node.